

FAST ITERATIVE SOLUTION OF REACTION-DIFFUSION CONTROL PROBLEMS ARISING FROM CHEMICAL PROCESSES*

JOHN W. PEARSON[†] AND MARTIN STOLL[‡]

Abstract. PDE-constrained optimization problems, and the development of preconditioned iterative methods for the efficient solution of the arising matrix systems, is a field of numerical analysis that has recently been attracting much attention. In this paper, we analyze and develop preconditioners for matrix systems that arise from the optimal control of reaction-diffusion equations, which themselves result from chemical processes. Important aspects of our solvers are saddle point theory, mass matrix representation, and effective Schur complement approximation, as well as the incorporation of control constraints and application of the outer (Newton) iteration to take into account the nonlinearity of the underlying PDEs.

Key words. PDE-constrained optimization, reaction-diffusion, chemical processes, Newton iteration, preconditioning, Schur complement

AMS subject classifications. 65F08, 65F10, 65F50, 92E20, 93C20

DOI. 10.1137/120892003

1. Introduction. A class of problems which has numerous applications within mathematical and physical problems is that of PDE-constrained optimization problems. One field in which these problems can be posed is that of chemical processes [4, 19, 20, 21, 22]. In this case the underlying PDEs are reaction-diffusion equations, and therefore the PDE constraints in our formulation are nonlinear PDEs.

When solving such reaction-diffusion control problems using a finite element method, and employing a Lagrange–Newton iteration to take account of the nonlinearity involved in the PDEs, the resulting matrix system for each Newton iteration will be large, sparse, and of saddle point structure. It is therefore desirable to devise preconditioned iterative methods to solve these systems efficiently and in such a way that the structure of the matrix is exploited. Work in constructing preconditioners for PDE-constrained optimization problems has been considered for simpler problems previously, for instance, Poisson control [46, 47, 53], convection-diffusion control [45], Stokes control [36, 50, 56], and heat equation control [44, 54].

In this paper, we will consider an optimal control formulation of a reaction-diffusion problem, which generates a symmetric matrix system upon each Newton iteration. (Such an iteration is required to take into account the nonlinear terms within the underlying PDEs.) We will generally search for block triangular preconditioners for the matrix systems we examine, to be used in conjunction with a suitable iterative solver. In order to do this, we will need to approximate the $(1, 1)$ -block by accurately representing the inverse of mass matrices amongst other things, as well as

*Submitted to the journal's Computational Methods in Science and Engineering section September 20, 2012; accepted for publication (in revised form) June 26, 2013; published electronically September 10, 2013. This work was supported by the award of a European Science Foundation (ESF) Exchange Grant under the OPTPDE program.

<http://www.siam.org/journals/sisc/35-5/89200.html>

[†]Numerical Analysis Group, Mathematical Institute, University of Oxford, 24–29 St Giles', Oxford, OX1 3LB, UK (john.pearson@worc.ox.ac.uk). This author's work was supported by the Engineering and Physical Sciences Research Council (UK), grant EP/P505216/1.

[‡]Computational Methods in Systems and Control Theory, Max Planck Institute for Dynamics of Complex Technical Systems, Sandtorstr. 1, 39106 Magdeburg, Germany (stollm@mpi-magdeburg.mpg.de).

devise an effective approximation of the Schur complement of the matrix system. We demonstrate with numerical tests why the choices we make are sensible for a number of practical problems.

This paper is structured as follows. In section 2, we discuss the underlying chemical problem (detailing the statement of the problem without and with control constraints included) and represent it in terms of matrix systems. In section 3, we introduce some basic saddle point theory and use this to devise effective preconditioners for the matrices which arise. In section 4, we present numerical results to demonstrate the performance of our iterative solvers in practice. Finally in section 5, we make some concluding remarks.

2. Problem formulation and discretization. Throughout this paper we consider an optimal control problem based on that considered in [4]. The objective function that has to be minimized is given by

$$(2.1) \quad J(u, v, c) = \frac{\alpha_u}{2} \|u - u_Q\|_{L_2(Q)}^2 + \frac{\alpha_v}{2} \|v - v_Q\|_{L_2(Q)}^2 + \frac{\alpha_{TU}}{2} \|u(\mathbf{x}, T) - u_\Omega\|_{L_2(\Omega)}^2 + \frac{\alpha_{TV}}{2} \|v(\mathbf{x}, T) - v_\Omega\|_{L_2(\Omega)}^2 + \frac{\alpha_c}{2} \|c\|_{L_2(\Sigma)}^2,$$

where u and v refer to concentrations of reactants (which in this problem are *state variables*), and c is the *control variable*, which also influences the underlying reaction. The spatial domain on which the problem is solved is given by $\Omega \subset \mathbb{R}^d$ with $d \in \{2, 3\}$, and the time domain is taken to be the interval $t \in [0, T]$. We then have the space-time domain Q given by $Q := \Omega \times [0, T]$, as well as the space-time boundary given by $\Sigma := \partial\Omega \times (0, T)$. The goal of the optimization problem is to compute the quantities u , v , and c in such a way that they are close in the L_2 -norm to what are often referred to as the *desired states* ($u_Q, v_Q, u_\Omega, v_\Omega$). Note that we have four desired states in this problem—two which are defined at all time points and two which are solely defined at the final time at which the problem is being solved. These are known quantities, which are typically determined from measurements and observations. In order for the objective function to resemble a physical or chemical process the variables need to satisfy the physics of the process of interest, which is typically modeled using one or more PDEs alongside additional constraints. In our case the constraints subject to which the objective function $J(u, v, c)$ is minimized are given by the following reaction-diffusion equations:

$$(2.2) \quad u_t - D_1 \Delta u + k_1 u = -\gamma_1 uv \quad \text{in } Q,$$

$$(2.3) \quad v_t - D_2 \Delta v + k_2 v = -\gamma_2 uv \quad \text{in } Q,$$

$$(2.4) \quad D_1 \partial_\nu u + b(\mathbf{x}, t, u) = c \quad \text{on } \Sigma,$$

$$(2.5) \quad D_2 \partial_\nu v + \tilde{\varepsilon} v = 0 \quad \text{on } \Sigma,$$

$$(2.6) \quad u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad \text{in } \Omega,$$

$$(2.7) \quad v(\mathbf{x}, 0) = v_0(\mathbf{x}) \quad \text{in } \Omega,$$

$$(2.8) \quad c \in C_{ad} = \{c \in L_\infty(\Sigma) : c_a \leq c \leq c_b \text{ a.e. on } \Sigma\}.$$

The quantities $\alpha_u, \alpha_v, \alpha_{TU}, \alpha_{TV}, \alpha_c, D_1, D_2, k_1, k_2, \gamma_1, \gamma_2$, and $\tilde{\varepsilon}$ are nonnegative constants. The function c describing the boundary condition (2.4) is the control variable defined above, and ∂_ν denotes the normal derivative. Equations (2.6) and (2.7) define the initial conditions for both concentrations. Additionally, we can impose

so-called box constraints on the control as stated in (2.8). In [22] Griesse and Volkwein also consider an integral constraint on c , which we do not discuss here. In some cases it might also be sensible to include state constraints for the concentrations u and v , which would be described by

$$u_a \leq u \leq u_b, \quad v_a \leq v \leq v_b.$$

State constraints typically bring additional difficulties to optimal control problems (see [10, 33]) and are not considered further in this present paper. For the remainder of this paper we will also follow the assumptions of $b(\mathbf{x}, t, u) = 0$ and $\tilde{\varepsilon} = 0$, as studied in [22]. There are two approaches for solving the above problem. The first is the so-called discretize-then-optimize approach, where we discretize the objective function and constraint to build a discrete Lagrangian, and then impose the optimality conditions in the discrete setting. The second is known as the optimize-then-discretize approach, where we instead build a Lagrangian for the infinite dimensional problem and then discretize the first order conditions. There is no preferred approach and we refer to [30] for a discussion of the two cases. We note that recently it has become a paradigm to create discretization schemes such that both approaches lead to the same discrete first order system. We also need to deal with the nonlinearity of the PDE constraint. We here apply a simple sequential quadratic programming (SQP) or Lagrange–Newton method. Before we proceed to the derivation of optimality conditions and discretization, we split the problem into two stages: solving the nonlinear PDEs without control constraints and solving the system with the additional control constraints incorporated.

Newton system without control constraints. In this section we wish to further describe how the above problem can be examined and in particular focus on how to treat the nonlinearity of the PDEs. We proceed by formally building the (continuous) Lagrangian subject to the reaction-diffusion system

$$\begin{aligned} u_t - D_1 \Delta u + k_1 u &= -\gamma_1 uv & \text{in } Q, \\ v_t - D_2 \Delta v + k_2 v &= -\gamma_2 uv & \text{in } Q, \\ D_1 \partial_\nu u &= c & \text{on } \Sigma, \\ D_2 \partial_\nu v &= 0 & \text{on } \Sigma, \\ u(\mathbf{x}, 0) &= u_0(\mathbf{x}) & \text{in } \Omega, \\ v(\mathbf{x}, 0) &= v_0(\mathbf{x}) & \text{in } \Omega, \end{aligned}$$

giving

$$\begin{aligned} \mathcal{L}(u, v, c, p, q) &= J(u, v, c) + \int_Q p(u_t - D_1 \Delta u + k_1 u + \gamma_1 uv) \\ &\quad + \int_Q q(v_t - D_2 \Delta v + k_2 v + \gamma_2 uv) \\ &\quad + \int_\Sigma p_\Sigma(D_1 \partial_\nu u - c) + \int_\Sigma q_\Sigma(D_2 \partial_\nu v). \end{aligned}$$

Here we have split up the *adjoint variables* p and q into interior and boundary parts (p and p_Σ , and q and q_Σ). We note that for brevity, when constructing \mathcal{L} , we included only the PDE part without boundary and initial conditions, which of course also need to be incorporated. We also make the assumption $\alpha_{TU} = \alpha_{TV} = 0$ in the working

below; the case where this is not so may be treated similarly. A rigorous derivation of the first order conditions can be found in [4, 22], to which we refer the interested reader. By taking the Fréchet derivatives with respect to state, control, and adjoint variables and equating the resulting expressions to zero, we obtain the first order conditions, or Karush–Kuhn–Tucker (KKT) conditions, given by

$$\begin{aligned}
 -p_t - D_1\Delta p + k_1p + \gamma_1pv + \gamma_2qv + \alpha_u(u - u_Q) &= 0 && \text{in } Q, \\
 -q_t - D_2\Delta q + k_2q + \gamma_2qu + \gamma_1pu + \alpha_v(v - v_Q) &= 0 && \text{in } Q, \\
 \partial_\nu p = \partial_\nu q &= 0 && \text{on } \Sigma, \\
 \alpha_c c - p &= 0 && \text{on } \Sigma, \\
 u_t - D_1\Delta u + k_1u + \gamma_1uv &= 0 && \text{in } Q, \\
 v_t - D_2\Delta v + k_2v + \gamma_2uv &= 0 && \text{in } Q, \\
 \partial_\nu u - D_1^{-1}c &= 0 && \text{on } \Sigma, \\
 \partial_\nu v &= 0 && \text{on } \Sigma.
 \end{aligned}$$

We may abbreviate this set of nonlinear equations describing the first order conditions, using the notation $\Phi(\mathbf{x}) = \mathbf{0}$. We can use Newton's method to solve this problem via the relation $\Phi'(\mathbf{x}_k)\mathbf{s}_k = -\Phi(\mathbf{x}_k)$.

We now construct the Fréchet derivative of Φ , obtaining

$$(2.9) \quad -(s_p)_t - D_1\Delta s_p + k_1s_p + \gamma_1(ps_v + s_pv) + \gamma_2(qs_v + s_qv) + \alpha_us_u = b_1,$$

$$(2.10) \quad -(s_q)_t - D_2\Delta s_q + k_2s_q + \gamma_2(qs_u + s_qu) + \gamma_1(ps_u + s_pu) + \alpha_vs_v = b_2,$$

$$(2.11) \quad \alpha_cs_c - s_p = b_3,$$

$$(2.12) \quad (s_u)_t - D_1\Delta s_u + k_1s_u + \gamma_1(vs_u + s_vu) = b_4,$$

$$(2.13) \quad (s_v)_t - D_2\Delta s_v + k_2s_v + \gamma_2(us_v + s_uv) = b_5.$$

Here we denote with $\mathbf{b} = [b_1, b_2, b_3, b_4, b_5]^T := -\Phi(\mathbf{x}_k)$ the right-hand side of the Newton system. Note that we did not write down the boundary conditions; however, they naturally carry through to the Newton system. If we now write all the equations together into an infinite dimensional system, the matrix describing the Newton process is given by

$$(2.14) \quad \begin{bmatrix} \alpha_u \text{Id} & \gamma_1p + \gamma_2q & 0 & \mathcal{L}'_u & \gamma_2v \\ \gamma_2q + \gamma_1p & \alpha_v \text{Id} & 0 & \gamma_1u & \mathcal{L}'_v \\ 0 & 0 & \alpha_c D_1^{-1} \text{Id} & -D_1^{-1} \text{Id} & 0 \\ \mathcal{L}_u & \gamma_1u & -D_1^{-1} \text{Id} & 0 & 0 \\ \gamma_2v & \mathcal{L}_v & 0 & 0 & 0 \end{bmatrix},$$

where

$$\begin{aligned}
 \mathcal{L}_u &= \frac{\partial}{\partial t} - D_1\Delta + k_1\text{Id} + \gamma_1v, & \mathcal{L}'_u &= -\frac{\partial}{\partial t} - D_1\Delta + k_1\text{Id} + \gamma_1v, \\
 \mathcal{L}_v &= \frac{\partial}{\partial t} - D_2\Delta + k_2\text{Id} + \gamma_2u, & \mathcal{L}'_v &= -\frac{\partial}{\partial t} - D_2\Delta + k_2\text{Id} + \gamma_2u,
 \end{aligned}$$

and Id denotes the identity operator.

In order to numerically solve the above problem we need to discretize the system (2.14) and the right-hand side $-\Phi(\mathbf{x}_k)$.

We first note that the system (2.14) is in saddle point form (as defined in section 3), and its discrete counterpart (using a backward Euler time-stepping scheme) is given by

$$(2.15) \quad \underbrace{\begin{bmatrix} \tau\mathcal{M}_1 & 0 & \mathcal{K}^T \\ 0 & \alpha_c\tau D_1^{-1}\mathcal{M}_c & -\tau D_1^{-1}\mathcal{N}^T \\ \mathcal{K} & -\tau D_1^{-1}\mathcal{N} & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} \mathbf{y} \\ \mathbf{c} \\ \boldsymbol{\lambda} \end{bmatrix} = \mathbf{b}$$

with

$$\begin{aligned} \mathcal{M}_1 &= \text{blkdiag} \left(M_1^{(1)}, M_1^{(2)}, \dots, M_1^{(N_t-1)}, M_1^{(N_t)} \right), \\ \mathcal{M}_c &= \text{blkdiag} (M_c, M_c, \dots, M_c, M_c), \\ \mathcal{N} &= \begin{bmatrix} N & & & & & \\ & 0 & & & & \\ & & N & & & \\ & & & 0 & & \\ & & & & \ddots & \\ & & & & & N \\ & & & & & & 0 \end{bmatrix}, \end{aligned}$$

where

$$M_1^{(i)} = \begin{bmatrix} \alpha_u M & \gamma_1 M_{p^{(i)}} + \gamma_2 M_{q^{(i)}} \\ \gamma_1 M_{p^{(i)}} + \gamma_2 M_{q^{(i)}} & \alpha_v M \end{bmatrix}, \quad i = 1, \dots, N_t.$$

Here, M denotes a standard finite element mass matrix, M_c is a boundary mass matrix, and the matrix N consists of evaluations of inner products from the term $\int_{\partial\Omega} w \text{tr}(z)$ with w a function on the boundary $\partial\Omega$, z a test function for the domain Ω , and tr the trace operator. The matrices $M_{p^{(i)}}$ and $M_{q^{(i)}}$ are mass-like matrices the entries of which are terms of the form $\int_{\Omega} \bar{p} \phi_j \phi_l$ and $\int_{\Omega} \bar{q} \phi_j \phi_l$, respectively (where \bar{p} and \bar{q} represent the previous Newton iterates of the adjoint variables—or Lagrange multipliers— p and q), and the vectors \mathbf{y} and $\boldsymbol{\lambda}$ correspond to the discretized state (\mathbf{u}, \mathbf{v}) and adjoint (\mathbf{p}, \mathbf{q}) variables, respectively. The quantity N_t denotes the number of time-steps used, with τ the size of the time-step.

Finally, the matrix \mathcal{K} represents the discretized PDE and can be written as

$$\mathcal{K} = \begin{bmatrix} L^{(1)} & & & & & \\ -M_d & L^{(2)} & & & & \\ & & \ddots & & & \\ & & & \ddots & & \\ & & & & -M_d & L^{(N_t-1)} \\ & & & & & -M_d & L^{(N_t)} \end{bmatrix},$$

where

$$M_d = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix}$$

and

$$L^{(i)} = \begin{bmatrix} M + \tau(D_1 K + k_1 M + \gamma_1 M_{v^{(i)}}) & \tau\gamma_1 M_{u^{(i)}} \\ \tau\gamma_2 M_{v^{(i)}} & M + \tau(D_2 K + k_2 M + \gamma_2 M_{u^{(i)}}) \end{bmatrix}$$

with K the standard finite element stiffness matrix, and $M_{u^{(i)}}$ and $M_{v^{(i)}}$ mass-like matrices with terms of the form $\int_{\Omega} \bar{u} \phi_j \phi_l$ and $\int_{\Omega} \bar{v} \phi_j \phi_l$, where \bar{u} and \bar{v} correspond to the previous Newton iterates of the state variables u and v .

Note that we can solve for the updated states, control, and adjoints directly, which also makes the computation of the right-hand side cheaper, that is,

$$\mathcal{A} \begin{bmatrix} u^{(k+1)} \\ v^{(k+1)} \\ c^{(k+1)} \\ p^{(k+1)} \\ q^{(k+1)} \end{bmatrix} = \mathcal{A} \begin{bmatrix} u^{(k)} \\ v^{(k)} \\ c^{(k)} \\ p^{(k)} \\ q^{(k)} \end{bmatrix} + \tilde{\mathbf{b}} = \begin{bmatrix} \alpha_u u_Q + (\gamma_1 p^{(k)} + \gamma_2 q^{(k)}) v^{(k)} \\ \alpha_v v_Q + (\gamma_2 q^{(k)} + \gamma_1 p^{(k)}) u^{(k)} \\ 0 \\ \gamma_1 u^{(k)} v^{(k)} \\ \gamma_2 v^{(k)} u^{(k)} \end{bmatrix},$$

in continuous form.

So far we have only discussed the Newton method to solve the KKT conditions. Note that for certain values of the states, Lagrange multipliers, and parameters we might run into the problem of obtaining an indefinite $(1, 1)$ -block of \mathcal{A} , caused by an indefinite matrix \mathcal{M}_1 [15]. For this reason we briefly highlight that for this purpose different techniques within the SQP step can be employed, such as line-search or trust region approaches—these may be explored in future research into this subject area. One alternative that we also mention within the numerical results of section 4 is a Gauss–Newton approach (see [24]), where we ignore all mixed derivatives of the Hessian with respect to the Lagrange multipliers, resulting in a matrix system defined by the matrix

$$(2.16) \quad \begin{bmatrix} \alpha_u \text{Id} & 0 & 0 & \mathcal{L}'_u & \gamma_2 v \\ 0 & \alpha_v \text{Id} & 0 & \gamma_1 u & \mathcal{L}'_v \\ 0 & 0 & \alpha_c D_1^{-1} \text{Id} & -D_1^{-1} \text{Id} & 0 \\ \mathcal{L}_u & \gamma_1 u & -D_1^{-1} \text{Id} & 0 & 0 \\ \gamma_2 v & \mathcal{L}_v & 0 & 0 & 0 \end{bmatrix}.$$

We find that preconditioners for the matrix (2.16) can be derived using the methodology presented in section 3.

Problem with control constraints. The problem we have discussed so far did not include any additional constraints on the control c . We now wish to discuss how pointwise constraints on the control, i.e.,

$$c_a(\mathbf{x}, t) \leq c(\mathbf{x}, t) \leq c_b(\mathbf{x}, t),$$

may be dealt with. The treatment of control constraints can typically be carried out using a semismooth Newton method introduced in [7]. (For further information we refer to [27, 30, 58].) For the special case of the reaction-diffusion system we point to literature such as [4, 19, 20, 21, 22] for discussions on control constraints. In general the gradient equation of the Lagrangian becomes a variational inequality, which is in turn solved using the semismooth Newton method or equivalently [27] a primal-dual active set method. In contrast to [7] we employ a penalty technique, which has been applied very successfully to state-constrained optimal control problems, called the Moreau–Yosida penalty function [25, 32, 37]—this approach has also been applied to control-constrained problems [55]. The advantage of this approach is that the method does not need to work on submatrices corresponding to the free variables, which would

require a reassembly of matrices for every Newton step and would make preconditioning the matrix systems more difficult. From experience [55], the performance of this approach is comparable to the approach that directly uses the semismooth Newton method. In the Moreau–Yosida framework, the constraints

$$c_a(\mathbf{x}, t) \leq c(\mathbf{x}, t) \leq c_b(\mathbf{x}, t)$$

are incorporated into the objective function via a penalization term, that is, we instead minimize the functional

$$J(u, v, c) + \frac{1}{2\varepsilon} \|\max\{0, c - c_b\}\|_{L_2(\Sigma)}^2 + \frac{1}{2\varepsilon} \|\min\{0, c - c_a\}\|_{L_2(\Sigma)}^2$$

subject to the state equations detailed above. We can now proceed using the semismooth Newton approach, solving linear systems of the form

$$(2.17) \quad \begin{bmatrix} \tau\mathcal{M}_1 & 0 & \mathcal{K}^T \\ 0 & \alpha_c\tau D_1^{-1}\mathcal{L}_c & -\tau D_1^{-1}\mathcal{N}^T \\ \mathcal{K} & -\tau D_1^{-1}\mathcal{N} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{c} \\ \boldsymbol{\lambda} \end{bmatrix} = \tilde{\mathbf{b}}$$

at each Newton step, where

$$\mathcal{L}_c = \begin{bmatrix} M_c + \varepsilon^{-1}G_{\mathcal{A}^{(1)}}M_cG_{\mathcal{A}^{(1)}} & & & \\ & \ddots & & \\ & & M_c + \varepsilon^{-1}G_{\mathcal{A}^{(N_t)}}M_cG_{\mathcal{A}^{(N_t)}} & \end{bmatrix}.$$

Here $\mathcal{A}^{(i)} = \mathcal{A}_+^{(i)} \cup \mathcal{A}_-^{(i)}$ defines the active sets for every time-step of the discretized problem, that is,

$$(2.18) \quad \mathcal{A}_+^{(i)} = \{j \in \{1, 2, \dots, N\} : (c_i)_j > (c_b)_{i,j}\},$$

$$(2.19) \quad \mathcal{A}_-^{(i)} = \{j \in \{1, 2, \dots, N\} : (c_i)_j < (c_a)_{i,j}\},$$

using the control \mathbf{c} from the previous iteration. The quantities $(c_i)_j$, $(c_b)_{i,j}$, and $(c_a)_{i,j}$ denote the values of c , c_b , and c_a at the i th time-step and the j th node, with N representing the total number of nodes. This method is schematically shown in Algorithm 1, where we assume here that the problem is already discretized.

ALGORITHM 1. Active set algorithm.

- 1: Choose initial values for $\mathbf{c}^{(0)}$, $\mathbf{p}^{(0)}$, $\mathbf{q}^{(0)}$, $\mathbf{u}^{(0)}$, $\mathbf{v}^{(0)}$
- 2: Set the active sets $\mathcal{A}_+^{(0)}$, $\mathcal{A}_-^{(0)}$ and $\mathcal{A}_I^{(0)}$ by using $\mathbf{c}^{(0)}$ in (2.18), (2.19)
- 3: **for** $k = 1, 2, \dots$ **do**
- 4: Solve (2.17) (a system on the free variables from the previous iteration ($\mathcal{A}_I^{(k-1)}$))
- 5: Set the active sets $\mathcal{A}_+^{(k)}$, $\mathcal{A}_-^{(k)}$ and $\mathcal{A}_I^{(k)}$ by using $\mathbf{c}^{(k)}$ as given in (2.18), (2.19)
- 6: **if** $\mathcal{A}_+^{(k)} = \mathcal{A}_+^{(k-1)}$, $\mathcal{A}_-^{(k)} = \mathcal{A}_-^{(k-1)}$, and $\mathcal{A}_I^{(k)} = \mathcal{A}_I^{(k-1)}$ **then**
- 7: STOP (Algorithm converged)
- 8: **end if**
- 9: **end for**

3. Solving the linear systems.

Krylov solvers. We now wish to discuss how to efficiently solve linear systems of the form $\mathcal{A}\mathbf{x} = \mathbf{b}$ that arise at the heart of the Lagrange–Newton method discussed in the previous section. We here decide to employ Krylov subspace methods, which have previously been found to be very efficient for a number of optimal control problems subject to PDE constraints [47, 48, 49, 53]. In our case, as the system matrix is symmetric and indefinite, one option would be to employ the MINRES [43] method introduced by Paige and Saunders. This is a short-term recurrence method [12], requiring only a minimal amount of storage and involving one matrix–vector multiplication per iteration. MINRES minimizes the 2-norm of the residual $\mathbf{r}_k = \mathbf{b} - \mathcal{A}\mathbf{x}_k$ over the current Krylov subspace, where \mathbf{x}_k is the approximation to \mathbf{x} at step k of this procedure. Alternatively, there are many widely used nonsymmetric solvers such as GMRES [52] and biconjugate gradients (BICG) [13] which could be used. Of course, any Krylov method should only be effective if a preconditioner \mathcal{P} is introduced such that the properties of the left-preconditioned system

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}$$

are better than that of the unpreconditioned system $\mathcal{A}\mathbf{x} = \mathbf{b}$. Specifically, \mathcal{P} is constructed in order to capture the properties of the matrix \mathcal{A} well and so that it is easy to invert. For excellent introductions to the topic of constructing preconditioners for saddle point problems, we refer to [5, 11] and the references mentioned therein. As a guideline for constructing good preconditioners we use the known results that if the saddle point matrix

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix}$$

is invertible, then the (ideal) block preconditioners

$$\mathcal{P}_1 = \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix}, \quad \mathcal{P}_2 = \begin{bmatrix} A & 0 \\ B & -S \end{bmatrix},$$

where A is the unchanged $(1, 1)$ -block of the saddle point matrix and $S = C + BA^{-1}B^T$ is the (negative) *Schur complement* of \mathcal{A} , satisfy $\lambda(\mathcal{P}_1^{-1}\mathcal{A}) \in \{1, \frac{1 \pm \sqrt{5}}{2}\}$ provided $C = 0$ [38, 39], and $\lambda(\mathcal{P}_2^{-1}\mathcal{A}) \in \{1\}$ for any matrix C [31]. Therefore, although \mathcal{P}_2 is nondiagonalizable, both \mathcal{P}_1 and \mathcal{P}_2 are extremely effective preconditioners for \mathcal{A} . Of course in practice, we would not wish to explicitly invert A and S to apply the ideal preconditioner; however, if we construct good approximations to the $(1, 1)$ -block and the Schur complement of the system (2.15), an appropriate iterative solver is likely to converge rapidly when used with a preconditioner consisting of these approximations. As pointed out earlier the $(1, 1)$ -block of the preconditioner may be indefinite—in this case we cannot employ a symmetric Krylov subspace solver. Now faced with the decision of choosing a nonsymmetric Krylov method, we wish to point out that it is not straightforward to pick the “best method” (see [40]) and the convergence of the Krylov subspace solver might not be adequately described by the eigenvalues of the preconditioned matrix system [18]. Nevertheless in practice a good clustering of the eigenvalues often leads to fast convergence of the iterative scheme, and it can be seen that with a good preconditioner many methods behave in a similar way.

It is also possible to employ multigrid approaches to such saddle point problems. This class of methods has previously been shown to demonstrate good performance when applied to solve a number of PDE-constrained optimization problems, subject to both steady and transient PDEs [1, 2, 8, 9, 23, 24, 28, 29, 57].

We emphasize once more that the matrix systems we seek to solve fit into this saddle point framework. For the problem described in section 2 without control constraints, for instance,

$$A = \begin{bmatrix} \tau\mathcal{M}_1 & 0 \\ 0 & \alpha_c\tau D_1^{-1}\mathcal{M}_c \end{bmatrix}, \quad B = [\mathcal{K} \quad -\tau D_1^{-1}\mathcal{N}], \quad C = [0].$$

We may therefore employ the theory of saddle point systems to develop preconditioners for this problem.

Approximating the (1, 1)-block. In the case of a PDE-constrained optimization problem with a linear PDE as the constraint and a cost functional of the form discussed in section 2, the (1, 1)-block of the resulting matrix system is a block diagonal matrix containing mass matrices (see [47, 49, 53], for instance), which can be handled very efficiently. In this case, however, we have to take into account that the (1, 1)-block now contains blocks of the form

$$(3.1) \quad \begin{bmatrix} \alpha_u M & \gamma_1 M_{p^{(i)}} + \gamma_2 M_{q^{(i)}} \\ \gamma_1 M_{p^{(i)}} + \gamma_2 M_{q^{(i)}} & \alpha_v M \end{bmatrix},$$

which demonstrates one of the major complexities encountered when attempting to solve such nonlinear problems numerically. When we seek to approximate these blocks, we use the saddle point theory as stated above to take as our approximation

$$\begin{bmatrix} \alpha_u M - \alpha_v^{-1} (\gamma_1 M_{p^{(i)}} + \gamma_2 M_{q^{(i)}}) M^{-1} (\gamma_1 M_{p^{(i)}} + \gamma_2 M_{q^{(i)}}) & 0 \\ \gamma_1 M_{p^{(i)}} + \gamma_2 M_{q^{(i)}} & \alpha_v M \end{bmatrix} \\ =: \begin{bmatrix} A_0^{(i)} & 0 \\ \gamma_1 M_{p^{(i)}} + \gamma_2 M_{q^{(i)}} & \alpha_v M \end{bmatrix}.$$

Using the saddle point result concerning block triangular preconditioners, we observe that each preconditioned block has eigenvalues all equal to 1 using this approximation. Note that these complicated looking matrices are actually straightforward to handle as we assume that the mass matrices are lumped for our work.¹ The block \mathcal{M}_c , which also forms part of the (1, 1)-block of our matrix systems, may be approximated using Chebyshev semi-iteration [16, 17, 59] for consistent mass matrices or by simple inversion for lumped mass matrices.

Approximating the Schur complement. We now focus on approximating the Schur complement of the matrix system, which is given by

$$S = \frac{1}{\tau} \mathcal{K} \mathcal{M}_1^{-1} \mathcal{K}^T + \frac{\tau}{\alpha_c D_1} \mathcal{N} \mathcal{M}_c^{-1} \mathcal{N}^T.$$

One approach that we would predict to prove successful for moderate values of the parameter α_c (motivated by work undertaken in [47], for instance) is to use the approximation

$$(3.2) \quad \hat{S}_1 = \frac{1}{\tau} \mathcal{K} \mathcal{M}_1^{-1} \mathcal{K}^T,$$

¹In the case where mass matrices are not lumped, we believe that we may take a similar approximation but replace the mass matrices by their diagonals within the preconditioner.

that is, to drop the second term of the exact Schur complement for our approximation. However for smaller values of α_c we find this approximation does not produce satisfactory results. Hence, approximations that provide robustness with respect to the crucial problem parameters have been investigated (see [35, 44, 46, 53, 60]). The idea introduced in [44, 46] for simpler problems uses an approximation of the form

$$\widehat{S}_2 = \frac{1}{\tau} (\mathcal{K} + \widehat{\mathcal{M}}) \mathcal{M}_1^{-1} (\mathcal{K} + \widehat{\mathcal{M}})^T,$$

where $\widehat{\mathcal{M}}$ is chosen to accommodate a better approximation of the term that was initially dropped from S . To discover an approach for finding such an approximation, we first study \widehat{S}_2 more closely:

$$\widehat{S}_2 = \frac{1}{\tau} (\mathcal{K} \mathcal{M}_1^{-1} \mathcal{K}^T + \widehat{\mathcal{M}} \mathcal{M}_1^{-1} \widehat{\mathcal{M}} + \mathcal{K} \mathcal{M}_1^{-1} \widehat{\mathcal{M}} + \widehat{\mathcal{M}} \mathcal{M}_1^{-1} \mathcal{K}^T).$$

Our goal is for the second term of the Schur complement approximation \widehat{S}_2 to accurately approximate the second term of the exact Schur complement S , that is,

$$(3.3) \quad \frac{1}{\tau} \widehat{\mathcal{M}} \mathcal{M}_1^{-1} \widehat{\mathcal{M}} \approx \frac{\tau}{\alpha_c D_1} \mathcal{N} \mathcal{M}_c^{-1} \mathcal{N}^T.$$

We now consider a block diagonal approximation $\widehat{\mathcal{M}}$ and recall the block structure of the other matrices involved. The most complex term which needs to be considered is the \mathcal{M}_1^{-1} term, which involves inverting 2×2 block matrices of the form (3.1). To carry out this task, we observe that, given suitable invertibility conditions, the inverse of a 2×2 block matrix $\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ may be expressed as

$$\begin{bmatrix} (A_{11} - A_{12} A_{22}^{-1} A_{21})^{-1} & -(A_{11} - A_{12} A_{22}^{-1} A_{21})^{-1} A_{12} A_{22}^{-1} \\ -(A_{22} - A_{21} A_{11}^{-1} A_{12})^{-1} A_{21} A_{11}^{-1} & (A_{22} - A_{21} A_{11}^{-1} A_{12})^{-1} \end{bmatrix},$$

which can be easily checked. We may use this expression to note that the problem of finding a suitable approximation (3.3) can be reduced to finding a matrix $\widehat{\mathcal{M}} = \text{blkdiag}(\widehat{M}_1^{(1)}, \widehat{M}_2^{(1)}, \widehat{M}_1^{(2)}, \widehat{M}_2^{(2)}, \dots, \widehat{M}_1^{(N_t)}, \widehat{M}_2^{(N_t)})$ such that

$$\begin{bmatrix} \tau^{-1} \widehat{M}_1^{(i)} A_0^{-(i)} \widehat{M}_1^{(i)} & 0 \\ 0 & \tau^{-1} \alpha_v^{-1} \widehat{M}_2^{(i)} M^{-1} \widehat{M}_2^{(i)} \end{bmatrix} \approx \begin{bmatrix} \tau \alpha_c^{-1} D_1^{-1} N M_c^{-1} N^T & 0 \\ 0 & 0 \end{bmatrix}$$

for $i = 1, \dots, N_t$, where $A_0^{-(i)} := (A_0^{(i)})^{-1}$.

We may therefore conclude that it is appropriate to take $\widehat{M}_2^{(1)} = \widehat{M}_2^{(2)} = \dots = \widehat{M}_2^{(N_t)} = 0$, with $\widehat{M}_1^{(i)}$ chosen such that

$$\frac{1}{\tau} \widehat{M}_1^{(i)} A_0^{-(i)} \widehat{M}_1^{(i)} \approx \frac{\tau}{\alpha_c D_1} M_\Gamma,$$

where $M_\Gamma := N M_c^{-1} N^T$. Given that the matrices $A_0^{-(i)}$ and M_Γ are diagonal, the above criterion will be satisfied if $\widehat{M}_1^{(i)}$ is a diagonal matrix, with diagonal entries given by

$$\widehat{m}_{1,jj}^{(i)} = \frac{\tau}{\sqrt{D_1} \alpha_c} |a_{0,jj}^{(i)}|^{1/2} m_{\Gamma,jj}^{1/2},$$

where $a_{0,jj}^{(i)}$ and $m_{\Gamma,jj}$ are the j th diagonal entries of $A_0^{(i)}$ and M_Γ , respectively. It would also be appropriate to make the selection

$$\widehat{m}_{1,jj}^{(i)} = \frac{\tau}{\sqrt{D_1\alpha_c}} h^{\frac{1}{2}(d-1)} |a_{0,jj}^{(i)}|^{1/2}$$

when j corresponds to a node on $\partial\Omega$, using the fact that M_c (which is equal to the nonzero part of M_Γ) is spectrally equivalent to $h^{d-1}I$, where d is the dimension of Ω and h the mesh-size used.

We may build these choices of $\widehat{M}_1^{(i)}$ into the matrix $\widehat{\mathcal{M}}$ and in turn into the Schur complement approximation \widehat{S}_2 . We can also check heuristically that these choices of $\widehat{\mathcal{M}}$ ensure that $\tau^{-1}\widehat{\mathcal{M}}\mathcal{M}_1^{-1}\widehat{\mathcal{M}} \approx \tau\alpha_c^{-1}D_1^{-1}\mathcal{N}\mathcal{M}_c^{-1}\mathcal{N}^T$ by taking the approximations $M \approx h^dI$, $M_c \approx h^{d-1}I$ (where I are identity matrices of different dimensions), and writing

$$\begin{aligned} \left(\frac{1}{\tau} \widehat{\mathcal{M}} \mathcal{M}_1^{-1} \widehat{\mathcal{M}} \right)_{jj} &\approx \frac{1}{\tau} \cdot \frac{\tau}{\sqrt{D_1\alpha_c}} h^{\frac{1}{2}(d-1)} |a_{0,jj}^{(i)}|^{1/2} \cdot a_{0,jj}^{-(i)} \cdot \frac{\tau}{\sqrt{D_1\alpha_c}} h^{\frac{1}{2}(d-1)} |a_{0,jj}^{(i)}|^{1/2} \\ &= \frac{\tau}{\alpha_c D_1} h^{d-1} \approx \left(\frac{\tau}{\alpha_c D_1} \mathcal{N} \mathcal{M}_c^{-1} \mathcal{N}^T \right)_{jj} \end{aligned}$$

whenever the index j corresponds to a boundary node. (Both sides of the expression would be equal to zero otherwise.) In the above work, we have assumed that $a_{0,jj}^{(i)} \neq 0$.

Let us now consider how our approximations of the (1, 1)-block A and (negative) Schur complement S may be applied. Due to the potential indefiniteness of A , as well as the nonsymmetry of the preconditioner used, a nonsymmetric solver, such as GMRES or BICG, needs to be applied. Given that this is the case, we recommend that a block triangular preconditioner of the form \mathcal{P}_2 be used, of the following structure:

$$\mathcal{P}_2 = \begin{bmatrix} \tau \widehat{\mathcal{M}}_1 & 0 & 0 \\ 0 & \alpha_c \tau D_1^{-1} \mathcal{M}_c & 0 \\ \mathcal{K} & -\tau D_1^{-1} \mathcal{N} & -\widehat{S}_2 \end{bmatrix},$$

where $\widehat{\mathcal{M}}_1$ denotes the approximation of \mathcal{M}_1 described above.

Alternative Schur complement approximation. We note at this point that as we apply a nonsymmetric iterative method to solve the matrix system discussed, we see no reason a nonsymmetric Schur complement approximation could not be used. For instance, it seems feasible to utilize an approximation

$$\widehat{S}_3 = \frac{1}{\tau} (\mathcal{K} + \widehat{\mathcal{M}}) \mathcal{M}_1^{-1} (\mathcal{K} + \widehat{\mathcal{M}})^T,$$

where in general $\widehat{\mathcal{M}}$ is not equal to $\widetilde{\mathcal{M}}$. We may consider block diagonal matrices

$$\begin{aligned} \widehat{\mathcal{M}} &= \text{blkdiag} \left(\widehat{M}_{11}^{(1)}, 0, \widehat{M}_{11}^{(2)}, 0, \dots, \widehat{M}_{11}^{(N_t)}, 0 \right), \\ \widetilde{\mathcal{M}} &= \text{blkdiag} \left(\widetilde{M}_{21}^{(1)}, 0, \widetilde{M}_{21}^{(2)}, 0, \dots, \widetilde{M}_{21}^{(N_t)}, 0 \right) \end{aligned}$$

and, similarly to above, select $\widehat{M}_{11}^{(i)}$ and $\widetilde{M}_{21}^{(i)}$ to be diagonal matrices. Their diagonal entries may be given by

$$\begin{aligned} \widehat{m}_{11,jj}^{(i)} &= \frac{\tau}{\sqrt{D_1\alpha_c}} |a_{0,jj}^{(i)}|, \\ \widehat{m}_{21,jj}^{(i)} &= \frac{\tau}{\sqrt{D_1\alpha_c}} m_{\Gamma,jj}, \quad \text{or} \quad \widehat{m}_{21,jj}^{(i)} = \frac{\tau}{\sqrt{D_1\alpha_c}} h^{\frac{1}{2}(d-1)}, \end{aligned}$$

ALGORITHM 2. Inexact Uzawa method.

- 1: Select \mathbf{x}_0 .
- 2: **for** $k = 0, 1, 2, \dots$ **do**
- 3: $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathcal{P}^{-1} \left(\mathbf{b} - \left(L + \text{blkdiag}(\widehat{M}, 0) \right) \mathbf{x}_k \right)$
- 4: **end for**

for example. Such choices of \widehat{M} and \widetilde{M} should ensure that the approximation $\tau^{-1} \widehat{M} \mathcal{M}_1^{-1} \widetilde{M} \approx \tau \alpha_c^{-1} D_1^{-1} \mathcal{N} \mathcal{M}_c^{-1} \mathcal{N}^T$ holds, as for the Schur complement approximation \widehat{S}_2 .

We note that whether the Schur complement approximation \widehat{S}_2 or \widehat{S}_3 is used, the inverses of matrices of the form $\mathcal{K} + \widehat{M}$ need to be approximated for every application of the Schur complement. We here use a Uzawa scheme [51] that approximately solves for diagonal blocks of the form $L + \text{blkdiag}(\widehat{M}, 0)$ of $\mathcal{K} + \widehat{M}$. (See Algorithm 2 for a sketch of the routine used.) The preconditioner \mathcal{P} is of block diagonal form and for each of these matrices applies an algebraic multigrid (AMG) technique to approximate the diagonal blocks of $L + \text{blkdiag}(\widehat{M}, 0)$.

Preconditioning for Gauss–Newton system. Let us now consider whether the approach detailed above may be applied to the matrix systems arising from a Gauss–Newton method. The matrix involved is given by (2.16) in continuous form, which in discrete form results in the same matrix (2.15) as for the Newton method, except now with

$$\mathcal{M}_1 = \text{blkdiag}(\alpha_u M, \alpha_v M, \alpha_u M, \alpha_v M, \dots, \alpha_u M, \alpha_v M).$$

For this matrix, we may approximate the (1, 1)-block $A = \text{blkdiag}(\tau \mathcal{M}_1, \tau \alpha_c D_1^{-1} \mathcal{M}_c)$ exactly. When developing an approximation of the form

$$\widehat{S}_2 = \frac{1}{\tau} (\mathcal{K} + \widehat{M}) \mathcal{M}_1^{-1} (\mathcal{K} + \widehat{M})^T$$

to the Schur complement

$$S = \frac{1}{\tau} \mathcal{K} \mathcal{M}_1^{-1} \mathcal{K}^T + \frac{\tau}{\alpha_c D_1} \mathcal{N} \mathcal{M}_c^{-1} \mathcal{N}^T,$$

we may therefore look again for a matrix of the form \widehat{M} such that

$$\frac{1}{\tau} \widehat{M} \mathcal{M}_1^{-1} \widehat{M} \approx \frac{\tau}{\alpha_c D_1} \mathcal{N} \mathcal{M}_c^{-1} \mathcal{N}^T.$$

As for the Newton system, this problem reduces to finding an alternating block diagonal matrix $\widehat{M} = \text{blkdiag}(\widehat{M}_1^{(1)}, 0, \widehat{M}_1^{(2)}, 0, \dots, \widehat{M}_1^{(N_t)}, 0)$ such that

$$\frac{1}{\tau} \begin{bmatrix} \widehat{M}_1^{(i)} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \alpha_u M & 0 \\ 0 & \alpha_v M \end{bmatrix}^{-1} \begin{bmatrix} \widehat{M}_1^{(i)} & 0 \\ 0 & 0 \end{bmatrix} \approx \frac{\tau}{\alpha_c D_1} \begin{bmatrix} \mathcal{N} \mathcal{M}_c^{-1} \mathcal{N}^T & 0 \\ 0 & 0 \end{bmatrix}$$

for $i = 1, \dots, N_t$. This suggests that we should take

$$\frac{1}{\tau \alpha_u} \widehat{M}_1^{(i)} M^{-1} \widehat{M}_1^{(i)} \approx \frac{\tau}{\alpha_c D_1} M_\Gamma,$$

which is achieved if $\widehat{M}_1^{(1)} = \dots = \widehat{M}_1^{(N_t)} = \widehat{M}_1$, where \widehat{M}_1 is a diagonal matrix with diagonal entries

$$\widehat{m}_{1,jj} = \tau \sqrt{\frac{\alpha_u}{D_1 \alpha_c}} m_{jj}^{1/2} m_{\Gamma,jj}^{1/2} \quad \text{or} \quad \tau \sqrt{\frac{\alpha_u}{D_1 \alpha_c}} h^{d-\frac{1}{2}}.$$

Here, m_{jj} and $m_{\Gamma,jj}$ denote the j th diagonal entries of M and M_Γ , respectively.

These approximations of A and S may be incorporated into a block diagonal or block triangular saddle point preconditioner. For the numerical results of section 4, we will once again consider block triangular preconditioners for matrix systems of this form.

Preconditioning for control constraints. We also wish to examine how the system (2.17), which incorporates inequality constraints on the control variable, may be preconditioned effectively. The (1,1)-block now contains the matrix $\alpha_c \tau D_1^{-1} \mathcal{L}_c$, which is a simple block diagonal matrix that can be treated in the same way as the (1,1)-block of the problem without control constraints. Approximating the Schur complement

$$S = \frac{1}{\tau} \mathcal{K} \mathcal{M}_1^{-1} \mathcal{K}^T + \frac{\tau}{\alpha_c D_1} \mathcal{N} \mathcal{L}_c^{-1} \mathcal{N}^T$$

is again the more challenging task. We now wish to use the technique employed earlier and write

$$\widehat{S}_2 = \frac{1}{\tau} (\mathcal{K} + \widehat{\mathcal{M}}) \mathcal{M}_1^{-1} (\mathcal{K} + \widehat{\mathcal{M}})^T,$$

where $\widehat{\mathcal{M}}$ is chosen such that

$$\frac{1}{\tau} \widehat{\mathcal{M}} \mathcal{M}_1^{-1} \widehat{\mathcal{M}} \approx \frac{\tau}{\alpha_c D_1} \mathcal{N} \mathcal{L}_c^{-1} \mathcal{N}^T.$$

Note that $\frac{\tau}{\alpha_c D_1} \mathcal{N} \mathcal{L}_c^{-1} \mathcal{N}^T$ is a block diagonal matrix with blocks of the form

$$\frac{\tau}{\alpha_c D_1} N (M_c + \varepsilon^{-1} G_{\mathcal{A}^{(i)}} M_c G_{\mathcal{A}^{(i)}})^{-1} N^T, \quad i = 1, \dots, N_t,$$

alternating with zero blocks. Hence, we see that $\widehat{\mathcal{M}}$ should again have an alternating block diagonal structure, that is, $\widehat{\mathcal{M}} = \text{blkdiag}(\widehat{M}_1^{(1)}, 0, \widehat{M}_1^{(2)}, 0, \dots, \widehat{M}_1^{(N_t)}, 0)$, as before.

Let us now employ the notation $l_{\Gamma,jj}^{(i)}$ for the diagonal entries of the matrix $N (M_c + \varepsilon^{-1} G_{\mathcal{A}^{(i)}} M_c G_{\mathcal{A}^{(i)}})^{-1} N^T$, which will be nonzero on the diagonals corresponding to boundary nodes and zero otherwise. Then, in complete analogy to the case without control constraints, we may motivate the following choice for the diagonal entries of $\widehat{M}_1^{(i)}$:

$$\widehat{m}_{1,jj}^{(i)} = \frac{\tau}{\sqrt{D_1 \alpha_c}} |a_{0,jj}^{(i)}|^{1/2} (l_{\Gamma,jj}^{(i)})^{1/2}.$$

These choices of $\widehat{M}_1^{(i)}$ may again be built into the matrix $\widehat{\mathcal{M}}$ and hence the Schur complement approximation \widehat{S}_2 .

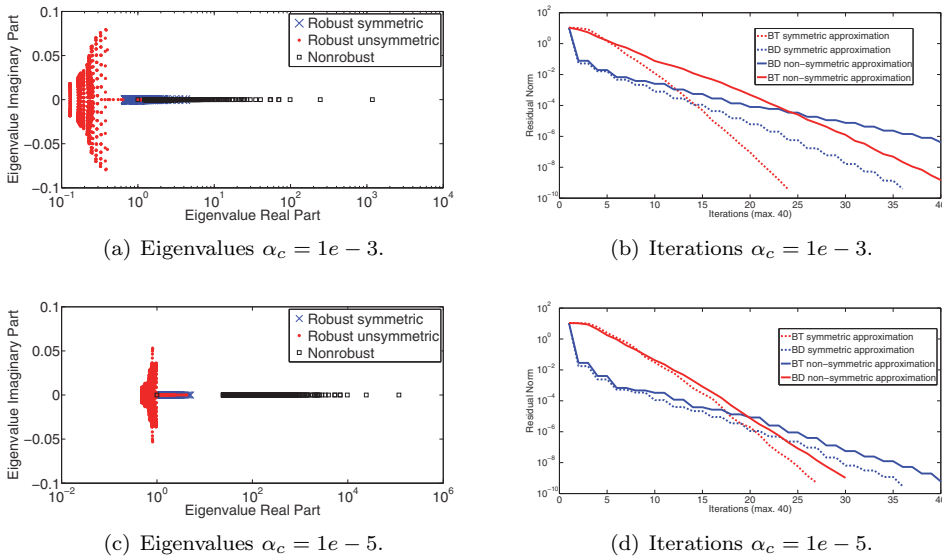


FIG. 3.1. Eigenvalues of $S\mathbf{v} = \lambda\widehat{S}\mathbf{v}$ for various approximations of the Schur complement (left) including a nonrobust approximation. GMRES iterations for the saddle point problem using the different Schur complement approximations and also block diagonal and block triangular preconditioners (right). The problem size is relatively small ($\dim(S) = 5000$).

Effectiveness of Schur complement approximations. To motivate our choices of Schur complement approximation, we wish to illustrate the properties of the preconditioned matrix systems when the approximations derived in this section are used. We note that good clustering of these eigenvalues alone will not guarantee rapid convergence of a nonsymmetric iterative solver such as BICG or GMRES; however, it should at least indicate the prudence of our selections.²

In Figure 3.1, we aim to demonstrate this effectiveness in two different ways. In plots (a) and (c) we show, for different values of α_c , the eigenvalues of the preconditioned Schur complement when robust symmetric (that is, \widehat{S}_2), robust unsymmetric (\widehat{S}_3), and nonrobust (\widehat{S}_1) approximations are used. The plots indicate that whereas the eigenvalues of both $\widehat{S}_2^{-1}S$ and $\widehat{S}_3^{-1}S$ seem to be fairly clustered, the results when \widehat{S}_2 is taken as the Schur complement approximation seem to be the best.

In plots (b) and (d), we show GMRES convergence plots for a test problem when a range of preconditioning choices are made, for different values of α_c . Specifically, we show results when the symmetric approximation (that is, \widehat{S}_2) and nonsymmetric approximation (\widehat{S}_3) are taken to the Schur complement and when a BT (block triangular preconditioner we considered in this section) and BD (analogous block diagonal preconditioner) are used within the GMRES method. Whereas the plots indicate that all four choices of preconditioner behave reasonably well, the block triangular preconditioner with Schur complement approximation \widehat{S}_2 yields the best results. We therefore use this preconditioner for the numerical results presented in the next section.

²Figure 3.1 considers the case without control constraints, but we note that the behavior of our preconditioners is very similar when control constraints are present.

4. Numerical experiments. In this section we present numerical results for the iterative methods we have described, using the Schur complement approximation \widehat{S}_2 . We implement this methodology using the finite element package deal.II [3] with Q_1 finite element basis functions on a quadrilateral mesh. For the AMG preconditioner, we use the Trilinos ML package [14] that applies a smoothed aggregation AMG. Within the multigrid routine we typically apply a Chebyshev smoother (10 steps) in combination with the application of 6 V-cycles. We currently regard our implementation as a proof of concept, as at present we reinitialize the AMG preconditioner upon each application. A possible alternative would be to store various preconditioners, however this is prohibitive from a computer memory point of view. Therefore, we believe that the development of an efficient technique using multigrid or a fixed number of steps of a simple iterative solver such as a Gauss–Seidel or Jacobi method should be investigated in the future. Consequently, we wish to emphasize that the timings presented here are not as rapid as they would be were the recomputation of the preconditioner at each application not required. If the varying preconditioners are handled efficiently, this could also lead to the relatively larger number of V-cycles being reduced—we choose to use this number of V-cycles as we wish to avoid the performance of the AMG routine being sensitive to parameter changes. Our implementation of BICG is stopped with a tolerance of 10^{-4} or smaller for the relative residual. Additionally, we stop the SQP method whenever the relative change between two consecutive solutions is smaller than a given tolerance, as specified in our examples. More sophisticated techniques [41] for carrying this out could be employed in the future. We feel that as our purpose is to illustrate the performance of our preconditioner the choice made here is appropriate. Our experiments are performed with $T = 1$ and $\tau = 0.05$, that is, with 20 time-steps. We take the parameters $\alpha_{TU} = \alpha_{TV} = 0$ in all our numerical experiments, though we find it makes little difference computationally if this is not the case. We only consider three-dimensional examples here and specify $\Omega \subset \mathbb{R}^3$ for each example. All results are performed on a Centos Linux machine with Intel Xeon CPU X5650 at 2.67 GHz CPUs and 48 GB of RAM. We present overall timings for the solution process in seconds.

No control constraints.

Example 1. The first example we consider involves a cylindrical shell domain shown in Figure 4.1(a) with inner radius 0.8, outer radius 1.0, and height 3.0. The parameter setup for this problem is as follows: the desired state for the first reactant is shown in Figure 4.1(b) and is given by

$$u_Q = 2t |\sin(2x_1x_2x_3)| + 0.3,$$

where $\mathbf{x} = [x_1, x_2, x_3]^T$, and the desired state for the second reactant is given by $v_Q = 0$. Additionally, we have $D_1 = D_2 = k_1 = k_2 = 1$ and $\gamma_1 = \gamma_2 = 0.15$. Figure 4.2 demonstrates computed state and control variables for this problem at a particular time-step.

In Table 4.1 we show for each step of the SQP method the number of BICG iterations needed to achieve the required convergence. The first column indicates the number of degrees of freedom (i.e., the dimension of the matrix systems being solved), with the second and fifth columns providing the timings for all SQP steps at the level of mesh refinement. The third and sixth columns give the number of SQP steps needed to reach convergence, and the remaining fourth and seventh columns give the iteration numbers needed for BICG to converge to the desired tolerance. For this setup we require three SQP steps to reach the tolerance of 10^{-6} . The results

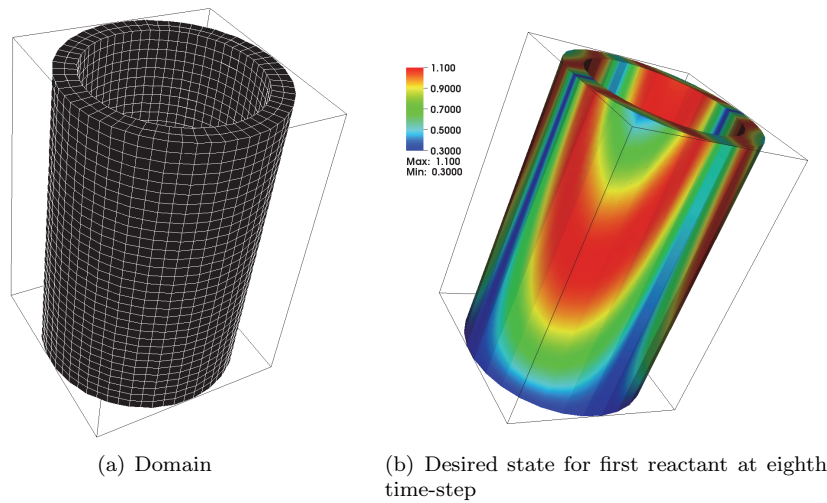


FIG. 4.1. Cylindrical shell domain for computations and desired state for the first reactant at the eighth time-step.

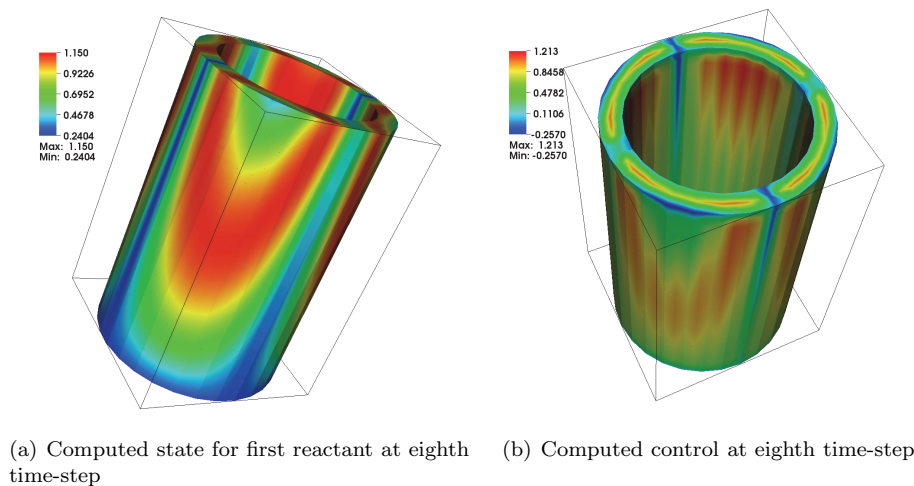


FIG. 4.2. Computed control and state for the first reactant at the eighth time-step with $\alpha_c = 1e - 5$ and $\alpha_u = \alpha_v = 1.0$.

indicate a benign mesh-dependence of the preconditioner, which from our experience can often be observed for boundary control problems. We also observe nearly constant iteration numbers when the regularization parameter is varied.

Example 2. The setup used for the second example is similar to that for the first. Here, however, we take the desired states

$$u_Q = \begin{cases} 0.7 & \text{for } (x_1, x_2, x_3) \in [0, 0.5]^3, \\ 0.2 & \text{otherwise,} \end{cases} \quad v_Q = 0,$$

with the parameters $D_1 = D_2 = k_1 = k_2 = 1$ and $\gamma_1 = \gamma_2 = 0.15$. In contrast to the previous example we solve the optimization problem on a Hyper L domain consisting

TABLE 4.1

Results on the cylindrical shell domain for varying mesh-size and regularization parameter α_c . SQP steps are shown in columns 3 and 6 with the corresponding BICG iteration numbers in columns 4 and 7. The timings (in seconds) measure the total time for convergence of the SQP scheme.

DoF	Time	SQP step	BICG	Time	SQP step	BICG	
		$\alpha_c = 1e-5$			$\alpha_c = 1e-3$		
538 240	1726	step 1	16	1995	step 1	17	
		step 2	16		step 2	20	
		step 3	16		step 3	20	
3 331 520	14904	step 1	28	14757	step 1	28	
		step 2	27		step 2	31	
		step 3	34		step 3	29	

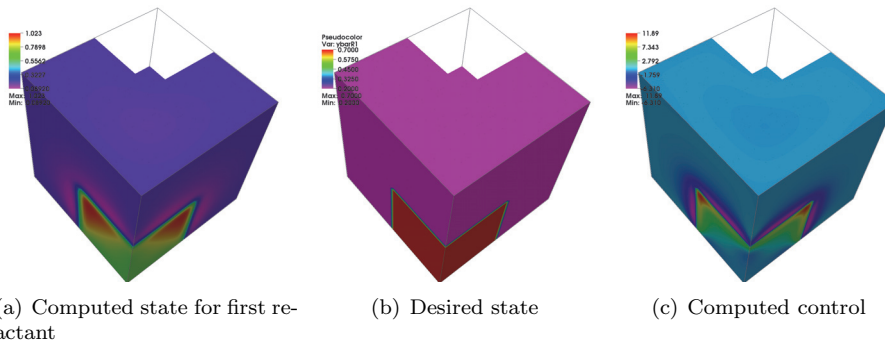


FIG. 4.3. Desired state, computed control, and state for the first reactant at eighth time-step with $\alpha_c = 1e-5$ and $\alpha_u = \alpha_v = 1.0$.

of the cube on $[-1, 1]^3$ with the cube $(0, 1]^3$ removed (see Figure 4.3). Again, we wish to vary the control regularization parameter α_c and the mesh-size. Table 4.2 shows the results for the setup presented here, including timings and iteration numbers as explained in Table 4.1. We again observe a mild growth in iteration numbers with varying mesh-size and robustness for our selection of α_c values. We find all iteration numbers are very reasonable considering the complexity of the matrix system being solved.

TABLE 4.2

Results on the Hyper L domain for varying mesh-size and regularization parameter α_c . SQP steps are shown in columns 3 and 6 with the corresponding BICG iteration numbers in columns 4 and 7. The timings (in seconds) measure the total time for convergence of the SQP scheme.

DoF	Time	SQP	BICG	Time	SQP	BICG	
		$\alpha_c = 1e-5$			$\alpha_c = 1e-3$		
60920	457	step 1	23	369	step 1	19	
		step 2	25		step 2	20	
		step 3	25		step 3	20	
382 840	2819	step 1	29	2624	step 1	27	
		step 2	35		step 2	30	
		step 3	33		step 3	33	
2 670 200	22976	step 1	46	19128	step 1	36	
		step 2	52		step 2	44	
		step 3	53		step 3	44	

Varying parameters. We next consider a problem where the desired states are given by

$$u_Q = \begin{cases} 0.3 & \text{for } (x_1, x_2, x_3) \in [0, 0.5]^3, \\ 0.2 & \text{otherwise,} \end{cases} \quad v_Q = 0,$$

and vary some values that have previously been assumed to be fixed. The default setup is again $D_1 = D_2 = k_1 = k_2 = 1$ and $\gamma_1 = \gamma_2 = 0.15$, with the stopping tolerance for the SQP method set as 10^{-4} . In the remainder of this section we vary one or two of these parameters and keep the others fixed. Clearly this does not cover all the relevant choices that might be possible, but this should indicate the effectiveness of our approach for a large range of parameter regimes. All computations are carried out on a fixed mesh that leads to a saddle point system of dimension 382840. We note that each of the problems tested represents a completely different setup of the PDE and the optimization problem. The purpose of presenting the results in Table 4.3 is to show that the iteration numbers for these scenarios are reasonable and sometimes very low. There are some specific parameter regimes for which this approach is not as effective as for the cases presented, but for a wide range of parameters ($h, \tau, \alpha_u, \alpha_v, \alpha_c, D_1, D_2, k_1, k_2, \gamma_1, \gamma_2$) we find that our approach works very well. Also presented in the table are results for the case when the tolerance of the iterative solver is decreased—it can be seen that the increase in iteration numbers is not dramatic for a decreased tolerance.

We may see from the results in Table 4.3 that especially for increasing values of γ_1 and γ_2 it is possible that the convergence deteriorates slightly due to the $(1, 1)$ -block having larger negative eigenvalues (as the increasing indefiniteness of the $(1, 1)$ -block in this case is not captured by our preconditioner). One way to overcome this issue is by switching from a Newton method to a Gauss–Newton scheme [6, 42]. This means that the off-diagonal blocks in (2.14) are ignored, which results in a typically slower convergence of the nonlinear iteration but provides better matrix properties during

TABLE 4.3

Results for varying parameters on the Hyper L domain with fixed dimension 382840 and varying regularization parameter α_c . The timings (in seconds) measure the total time for convergence of the SQP scheme. We show the number of BICG iterations and the number of SQP steps.

Parameter	Time	SQP step	BICG	Time	SQP step	BICG
		$\alpha_c = 1e - 5$			$\alpha_c = 1e - 3$	
$D_1 = D_2 = 100$	1783	step 1	28	1161	step 1	16
		step 2	33		step 2	22
$D_1 = D_2 = 0.1$	2083	step 1	19	1744	step 1	18
		step 2	27		step 2	20
		step 3	20		step 3	19
$\gamma_1 = \gamma_2 = 0.05$	2426	step 1	25	2199	step 1	22
		step 2	29		step 2	25
		step 3	29		step 3	25
$\gamma_1 = \gamma_2 = 0.75$	3240	step 1	20	3796	step 1	24
		step 2	60		step 2	36
		step 3	32		step 3	72
tol = $1e - 6$	3226	step 1	34	2702	step 1	27
		step 2	38		step 2	33
		step 3	38		step 3	33
tol = $1e - 8$	3749	step 1	39	3289	step 1	33
		step 2	46		step 2	39
		step 3	46		step 3	42

TABLE 4.4

Results for varying parameters γ_1 and γ_2 on the Hyper L domain with fixed dimension 382840 and varying regularization parameter α_c for the Gauss-Newton scheme. The timings (in seconds) measure the total time for convergence of the SQP scheme. We show the number of BICG iterations and the number of GN steps.

Parameter	Time	GN Step	BICG	Time	GN Step	BICG
		$\alpha_c = 1e-5$			$\alpha_c = 1e-3$	
$\gamma_1 = \gamma_2 = 0.75$	3107	step 1	25	2848	step 1	22
		step 2	27		step 2	25
		step 3	27		step 3	25
		step 4	27		step 4	24
$\gamma_1 = \gamma_2 = 1.75$	3249	step 1	25	2973	step 1	22
		step 2	28		step 2	26
		step 3	28		step 3	26
		step 4	26		step 4	26

the solution process. We observe that all our preconditioners can be applied in this case. Table 4.4 provides some results for this case with larger γ_1 and γ_2 and the desired states

$$u_Q = \begin{cases} 0.7 & \text{for } (x_1, x_2, x_3) \in [0, 0.5]^3, \\ 0.2 & \text{otherwise,} \end{cases} \quad v_Q = 0.$$

It can be seen that we have a small increase in the number of Gauss-Newton (GN in the table) iterations compared to the number of SQP steps to reach the tolerance of 10^{-4} but that the number of BICG iterations remains (almost) constant.

Control constraints. We now present results for the case where control constraints are present. The domain of interest is again the Hyper L domain presented earlier, with the desired states given by

$$u_Q = t |\sin(2x_1x_2x_3) \cos(2x_1x_2x_3)|, \quad v_Q = 0,$$

and $D_1 = D_2 = k_1 = k_2 = 1$, $\gamma_1 = \gamma_2 = 0.15$. We work only with an upper bound on the control given by

$$c_b = 1.5.$$

The results for varying α_c and different mesh-sizes are shown in Table 4.5. We note that the convergence of the Newton method dealing with the control constraints (CCNM in the tables) seems to depend on the tolerance used for the solution of

TABLE 4.5

Results on the Hyper L domain for varying mesh-size and regularization parameter α_c . Shown are the number of SQP steps, the number of the Newton iterations for the CCNM, and the average number of BICG iterations per step of the CCNM method.

DoF	Time	SQP step	CCNM/BICG	Time	SQP step	CCNM/BICG
		$\alpha_c = 1e-5$			$\alpha_c = 1e-3$	
60 920	859	step 1	3/22.0	1066	step 1	3/18.0
		step 2	2/25.5		step 2	3/21.0
					step 3	3/21.0
382 840	13358	step 1	5/28.6	5498	step 1	2/26.0
		step 2	5/32.6		step 2	2/36.0
		step 3	5/32.8		step 3	2/35.0

TABLE 4.6

Results on the Hyper L domain for varying penalty parameter ε . Shown are the number of SQP steps, the number of the Newton iterations for the CCNM, and the average number of BICG iterations per step of the CCNM method.

DoF	SQP step	CCNM/ av.BICG	SQP step	CCNM/ av.BICG	SQP step	CCNM/ av.BICG
		$\varepsilon = 1e-2$		$\varepsilon = 1e-4$		$\varepsilon = 1e-6$
60 920	step 1	3/32.3	step 1	3/24.6	step 1	3/20.6
	step 2	3/36.3	step 2	3/26.6	step 2	3/22.3

the linear system (see [34]). The smaller value of α_c shown in Table 4.5 requires the tolerance for the iterative solver to be reduced, as otherwise we do not observe convergence of the Newton method to deal with the control constraints. Our stopping criterion for the Newton method is based on the coincidence of subsequent active sets, but a more sophisticated stopping criterion might be able to avoid the convergence issue of the Newton method [26, 41]. Table 4.5 shows the number of SQP steps, the number of semismooth Newton steps (CCNM) for the control constraints, and the average number of BICG iterations at each SQP step. We find that it is also feasible to handle the nonlinearity of the PDEs and the control constraints within a single Newton loop, and the matrix systems obtained using this approach are of the same structure as that derived in section 2. We see that there is a benign growth in BICG iteration numbers with respect to the mesh-size. The difference in iteration numbers for the two different values of α_c is likely to be due to the fact that as we change α_c the values for the control c change, which means that more nodes belong to the active sets than in the case with the larger value of α_c .

In addition, we wish to illustrate robustness with respect to the penalty parameter ε . We here keep the mesh-size, as well as the regularization parameter ($\alpha_c = 1e-3$), fixed and consider different values of ε . Table 4.6 illustrates that once again the resulting iteration numbers are very reasonable given the complexity of the problem. We also observe that the performance of the Newton method depends on the tolerance with which the linear systems were solved. For the rather low tolerance of 10^{-9} we find that the Newton scheme and the SQP method often converge in very few iterations. We sometimes observe that for smaller values of the penalty parameter the convergence of the outer SQP method is slower than for larger values. This may be caused by the use of our simple SQP scheme—as we mentioned previously more sophisticated schemes may be able to avoid this. Observe that the residual of the iterative solver depends on ε , and thus from our experience small tolerances are typically required to ensure convergence of the outer iteration. We note that it is also possible to replace the SQP scheme by a Gauss–Newton iteration to possibly avoid indefinite Hessians.

5. Conclusions. In this paper we have considered a PDE-constrained optimization problem based on reaction-diffusion equations used to model chemical processes. We devised nonlinear solvers to solve these problems, at the heart of which lay the solution of large-scale linear systems of saddle point structure. We have shown that these systems can be solved using efficient preconditioning techniques for a wide range of cases. We have introduced a preconditioner that efficiently approximates the $(1, 1)$ -block of the saddle point systems, and we additionally derived approximations of the Schur complement which were intended to be robust with respect to parameters within the construction of the problem. Our numerical results illustrated that for a variety

of problem setups (including problems with box constraints on the control variable) our method solves the matrix systems in low BICG iteration numbers. To summarize, the method presented here not only enables the accurate solution of chemical process models but also provides fast and robust techniques to do this.

Acknowledgments. The authors gratefully acknowledge the Max Planck Institute in Magdeburg for their hospitality. The authors would like to thank Andy Wathen for fruitful discussions regarding the presented work.

REFERENCES

- [1] S. S. ADAVANI AND G. BIROS, *Multigrid algorithms for inverse problems with linear parabolic PDE constraints*, SIAM J. Sci. Comput., 31 (2008), pp. 369–397.
- [2] U. ASCHER AND E. HABER, *A multigrid method for distributed parameter estimation problems*, Electron. Trans. Numer. Anal., 15 (2003), pp. 1–17.
- [3] W. BANGERTH, R. HARTMANN, AND G. KANSCHAT, *deal.II. A general-purpose object-oriented finite element library*, ACM Trans. Math. Software, 33 (2007), pp. 24:1–24:7.
- [4] W. BARTHEL, C. JOHN, AND F. TRÖLTZSCH, *Optimal boundary control of a system of reaction diffusion equations*, ZAMM Z. Angew. Math. Mech., 90 (2010), pp. 966–982.
- [5] M. BENZI, G. H. GOLUB, AND J. LIESEN, *Numerical solution of saddle point problems*, Acta Numer., 14 (2005), pp. 1–137.
- [6] M. BENZI, E. HABER, AND L. TARALLI, *A preconditioning technique for a class of PDE-constrained optimization problems*, Adv. Comput. Math., 35 (2011), pp. 149–173.
- [7] M. BERGOUNIOUX, K. ITO, AND K. KUNISCH, *Primal-dual strategy for constrained optimal control problems*, SIAM J. Control Optim., 37 (1999), pp. 1176–1194.
- [8] A. BORZÌ, *Multigrid methods for parabolic distributed optimal control problems*, J. Comput. Appl. Math., 157 (2003), pp. 365–382.
- [9] A. BORZÌ AND V. SCHULZ, *Multigrid methods for PDE optimization*, SIAM Rev., 51 (2009), pp. 361–395.
- [10] E. CASAS, *Control of an elliptic problem with pointwise state constraints*, SIAM J. Control Optim., 24 (1986), pp. 1309–1318.
- [11] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics*, in Numerical Mathematics and Scientific Computation, Oxford University Press, New York, 2005.
- [12] V. FABER AND T. MANTEUFFEL, *Necessary and sufficient conditions for the existence of a conjugate gradient method*, SIAM J. Numer. Anal., 21 (1984), pp. 352–362.
- [13] R. FLETCHER, *Conjugate gradient methods for indefinite systems*, in Numerical Analysis, Lecture Notes in Math., 506, Springer, Berlin, 1976, pp. 73–89.
- [14] M. GEE, C. SIEFERT, J. HU, R. TUMINARO, AND M. SALA, *ML 5.0 Smoothed Aggregation User's Guide*, Tech. report SAND2006-2649, Sandia National Laboratories, 2006.
- [15] P. E. GILL AND W. MURRAY, *Newton-type methods for unconstrained and linearly constrained optimization*, Math. Program., 7 (1974), pp. 311–350.
- [16] G. H. GOLUB AND R. S. VARGA, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. I*, Numer. Math., 3 (1961), pp. 147–156.
- [17] G. H. GOLUB AND R. S. VARGA, *Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. II*, Numer. Math., 3 (1961), pp. 157–168.
- [18] A. GREENBAUM, V. PTÁK, AND Z. STRAKOŠ, *Any nonincreasing convergence curve is possible for GMRES*, SIAM J. Matrix Anal. Appl. 17 (1996), pp. 465–470.
- [19] R. GRIESSE, *Parametric sensitivity analysis in optimal control of a reaction diffusion system. I: Solution differentiability*, Numer. Funct. Anal. Optim., 25 (2004), pp. 93–117.
- [20] R. GRIESSE, *Parametric sensitivity analysis in optimal control of a reaction diffusion system. II: Practical methods and examples*, Optim. Methods Softw., 19 (2004), pp. 217–242.
- [21] R. GRIESSE AND S. VOLKWEIN, *A primal-dual active set strategy for optimal boundary control of a nonlinear reaction-diffusion system*, SIAM J. Control Optim., 44 (2005), pp. 467–494.
- [22] R. GRIESSE AND S. VOLKWEIN, *Parametric Sensitivity Analysis for Optimal Boundary Control of a 3D Reaction-Diffusion System*, Springer, New York, 2006.
- [23] E. HABER, *A parallel method for large scale time domain electromagnetic inverse problems*, Appl. Numer. Math., 58 (2008), pp. 422–434.

- [24] E. HABER, U. M. ASCHER, AND D. OLDENBURG, *On optimization techniques for solving nonlinear inverse problems*, *Inverse Problems*, 16 (2000), pp. 1263–1280.
- [25] R. HERZOG AND E. W. SACHS, *Preconditioned conjugate gradient method for optimal control problems with control and state constraints*, *SIAM J. Matrix Anal. Appl.*, 31 (2010), pp. 2291–2317.
- [26] M. HINTERMÜLLER, M. HINZE, AND M. TBER, *An adaptive finite-element Moreau-Yosida-based solver for a non-smooth Cahn-Hilliard problem*, *Optim. Methods Softw.*, 26 (2011), pp. 777–811.
- [27] M. HINTERMÜLLER, K. ITO, AND K. KUNISCH, *The primal-dual active set strategy as a semismooth Newton method*, *SIAM J. Optim.*, 13 (2002), pp. 865–888.
- [28] M. HINZE, M. KÖSTER, AND S. TUREK, *A Hierarchical Space-Time Solver for Distributed Control of the Stokes Equation*, Tech. report SPP1253-16-01, Priority Programme 1253, 2008.
- [29] M. HINZE, M. KÖSTER, AND S. TUREK, *A Space-Time Multigrid Solver for Distributed Control of the Time-Dependent Navier-Stokes System*, Tech. report, SPP1253-16-02, Priority Programme 1253, 2008.
- [30] M. HINZE, R. PINNAU, M. ULBRICH, AND S. ULBRICH, *Optimization with PDE Constraints*, in *Mathematical Modelling: Theory and Applications*, Springer, New York, 2009.
- [31] I. IPSEN, *A note on preconditioning non-symmetric matrices*, *SIAM J. Sci. Comput.*, 23 (2001), pp. 1050–1051.
- [32] K. ITO AND K. KUNISCH, *The primal-dual active set method for nonlinear optimal control problems with bilateral constraints*, *SIAM J. Control Optim.*, 43 (2004), pp. 357–376.
- [33] K. ITO AND K. KUNISCH, *Lagrange Multiplier Approach to Variational Problems and Applications*, *Adv. Des. Control* 15, SIAM, Philadelphia, 2008.
- [34] C. KANZOW, *Inexact semismooth Newton methods for large-scale complementarity problems*, *Optim. Methods Softw.*, 19 (2004), pp. 309–325.
- [35] M. KOLLMANN AND M. KOLMBAUER, *A Preconditioned MinRes Solver for Time-Periodic Parabolic Optimal Control Problems*, *Numer. Linear Algebra Appl.*, DOI: 10.1002/nla.1842, 2012.
- [36] M. KOLLMANN AND W. ZULEHNER, *A Robust Preconditioner for Distributed Optimal Control for Stokes Flow with Control Constraints*, *Numer. Math. Adv. Appl.*, (2011), pp. 771–779.
- [37] K. KRUMBIEGEL, I. NEITZEL, AND A. RÖSCH, *Sufficient Optimality Conditions for the Moreau-Yosida-Type Regularization Concept Applied to the Semilinear Elliptic Optimal Control Problems with Pointwise State Constraints*, Tech. report 1503/2010, *Weierstrass Institute for Applied Analysis and Stochastics (Berlin)*, 2010.
- [38] Y. A. KUZNETSOV, *Efficient iterative solvers for elliptic finite element problems on nonmatching grids*, *Russian J. Numer. Anal. Math. Modelling*, 10 (1995), pp. 187–211.
- [39] M. F. MURPHY, G. H. GOLUB, AND A. J. WATHEN, *A note on preconditioning for indefinite linear systems*, *SIAM J. Sci. Comput.*, 21 (2000), pp. 1969–1972.
- [40] N. M. NACHTIGAL, S. C. REDDY, AND L. N. TREFETHEN, *How fast are nonsymmetric matrix iterations?*, *SIAM J. Matrix Anal. Appl.*, 13 (1992), pp. 778–795.
- [41] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer Ser. Oper. Res., Springer, New York, 1999.
- [42] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer Ser. Oper. Res. Financial Engineering, 2nd ed., Springer, New York, 2006.
- [43] C. C. PAIGE AND M. A. SAUNDERS, *Solutions of sparse indefinite systems of linear equations*, *SIAM J. Numer. Anal.*, 12 (1975), pp. 617–629.
- [44] J. W. PEARSON, M. STOLL, AND A. J. WATHEN, *Regularization-robust preconditioners for time-dependent PDE-constrained optimization problems*, *SIAM J. Matrix Anal. Appl.*, 33 (2012), pp. 1126–1152.
- [45] J. W. PEARSON AND A. J. WATHEN, *Fast Iterative Solvers for Convection-Diffusion Control Problems*, *Electron. Trans. Numer. Anal.*, 40, (2013), pp. 294–310.
- [46] J. W. PEARSON AND A. J. WATHEN, *A new approximation of the Schur complement in preconditioners for PDE-constrained optimization*, *Numer. Linear Algebra Appl.*, 19 (2012), pp. 816–829.
- [47] T. REES, H. S. DOLLAR, AND A. J. WATHEN, *Optimal solvers for PDE-constrained optimization*, *SIAM J. Sci. Comput.*, 32 (2010), pp. 271–298.
- [48] T. REES AND M. STOLL, *Block-triangular preconditioners for PDE-constrained optimization*, *Numer. Linear Algebra Appl.*, 17 (2010), pp. 977–996.
- [49] T. REES, M. STOLL, AND A. WATHEN, *All-at-once preconditioners for PDE-constrained optimization*, *Kybernetika*, 46 (2010), pp. 341–360.

- [50] T. REES AND A. WATHEN, *Preconditioning iterative methods for the optimal control of the Stokes equation*, SIAM J. Sci. Comput., 33 (2011), pp. 2903–2926.
- [51] Y. SAAD, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, 2003.
- [52] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
- [53] J. SCHÖBERL AND W. ZULEHNER, *Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems*, SIAM J. Matrix Anal. Appl., 29 (2007), pp. 752–773.
- [54] M. STOLL AND A. WATHEN, *All-at-once solution of time-dependent PDE-constrained optimization problems*, Oxford Centre for Collaborative and Applied Mathematics Technical Report 10/47, 2010.
- [55] M. STOLL AND A. WATHEN, *Preconditioning for partial differential equation constrained optimization with control constraints*, Numer. Linear Algebra Appl., 19 (2012), pp. 53–71.
- [56] M. STOLL AND A. WATHEN, *All-at-once solution of time-dependent Stokes control*, J. Comput. Phys., 232 (2013), pp. 498–515.
- [57] S. TAKACS AND W. ZULEHNER, *Convergence analysis of multigrid methods with collective point smoothers for optimal control problems*, Comput. Vis. Sci., 14 (2011), pp. 131–141.
- [58] M. ULBRICH, *Semismooth Newton Methods for Variational Inequalities and Constrained Optimization Problems*, SIAM, Philadelphia, 2011.
- [59] A. J. WATHEN AND T. REES, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Electron. Trans. Numer. Anal., 34 (2008), pp. 125–135.
- [60] W. ZULEHNER, *Nonstandard norms and robust estimates for saddle point problems*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 536–560.